IN THE SPECIFICATION

Replace the "ABSTRACT OF THE DISCLOSURE" with the following paragraph:

A technique for observability based coverage of a design under test (DUT) is presented. A conventional simulation signal is augmented to include a "tag value." In the course of a simulation, assignment statements (for which observability-based coverage is desired) "inject" tag values on their output signals. A tag value contains an identifier uniquely identifying the assignment statement that produced it. A tag value also contains a "tag history." The tag history contains copies of the tag values for assignment statements earlier in the flow of control or in the flow of data. If a tag propagated through the DUT appears at an observable output, the circuit designer knows that the assignment statements it identifies have satisfied observability based coverage.

Replace the paragraph, from page 20, line 15 to page 21, line 2, with the following paragraph:

This rule-based approach to determining which tag values on the rhs of an assignment are propagated to the lhs is illustrated by the example of Figures 3D - 3F. Figure 3D depicts an example assignment statement which has two levels of Boolean operators. Figure 3E shows a parse tree of the rhs of the assignment of Figure 3D. As can be seen, each leaf node of the parse tree has a logical value and a tag value. For example, leaf node 340 has a logical value of 0 and a tag value whose tag ID portion 347 indicates it was injected by a particular assignment statement that set the value of the unique signal represented by "in_3." Leaf node 341 has a logical value of 1. The result of logical AND 342, as indicated by Figure 3E, is 0. Since only input "in_3" of leaf node 340 controlled the logical value for 342, only a copy of the tag value for node 340 is propagated to be the tag value for node intermediate node 342. With regard to the tag values for node 345, the tag values for both of its leaf nodes are included in its tag subset since both are controlling. With regard to root node 346, only copies of the tags of node 345 propagate to ~~become~~ become part of its tag subset since only the logical value of node 345 controls the logical value of node 346. Figure 3F depicts the tag value constructed for the lhs signal "out_1" of the assignment of Figure 3D.

Replace the paragraph, on page 21, lines 13-19, with the following paragraph:

In particular, consider the abstracted Verilog HDL code fragment of

Figure 4. Here, assignment statement 302 has been placed in a control-flow context of two nested “if” statements, both of which must be satisfied in order for assignment 302 to be executed. Just as with function f of assignment statement 302, function f_1 of outer “if” 440[340] and function f_2 of inner “if” 441[341] must each be evaluated for their subset of signals which have an observable effect on whether the function is satisfied.

Replace the paragraph, on page 37, lines 15-25, with the following paragraph:

Specifically, tag values whose tag ID, or whose tag history, contain the value L are located. Among the tag values found for a particular line number L, the objective is generally to locate those few tag values which represent furthest traversals along particular paths through the circuit design. Each such tag value, which represents a furthest traversal along a circuit path, shall be referred to as a “maximal traversal tag value.” These maximal traversal tag values are useful since they provide ~~they provide~~ information regarding fullest extent to which a value produced by an assignment L has traversed along a particular circuit path. For each such maximal traversal tag value, the assignment statement responsible for blocking L’s further traversal is identified from its tag last used value.